Information Knowledge Systems Management 5 (2005/2006) 189–209
IOS Press

Method engineering in practice: A case of applying the Zachman framework in the context of small enterprise architecture oriented projects

Tanja Ylimäki[a,*] and Veikko Halttunen[b]
[a]*Information Technology Research Institute, P.O. Box 35, FIN-40014 University of Jyväskylä, Finland*
[b]*Department of Computer Science and Information Systems, P.O. Box 35, FIN-40014 University of Jyväskylä, Finland*

Abstract: During the past few years, enterprise architectures (EA) have garnered considerable attention from the community of information systems (IS) practitioners and academics. It is suggested that EA is an approach for controlling the complexity and constant changes in the business environment of an organization, enabling a real alignment between the business vision, business requirements and information systems. Managing enterprise architectures is, however, a difficult and very complex task. Fortunately, a few frameworks have been developed to help understanding this complicated field: of these, the best known is the Zachman framework. In spite of its popularity, it is, nonetheless, hard to find scientific studies on applying or utilizing the Zachman framework. In this paper, we present a case study in which the Zachman framework was applied in a method engineering effort, aiming at improving method adoption in the context of small EA-oriented projects. In order to discuss the usability of the Zachman framework as a mapping tool, the development process and its outcome are described.

Keywords: Enterprise architecture, Zachman framework, method engineering, method adoption, small scale project

1. Introduction

During the past few years, enterprise architectures (EA) have garnered considerable attention from the community of IS practitioners and academics. It has been suggested that EA is an approach for controlling the complexity and constant changes in the business environment of an organization, enabling a true alignment between the business vision, business requirements and information systems [3,4,27]. Enterprise architectures are generally seen as blueprints that identify the essential parts of an organization (such as people, business processes, technology, information, financial elements, and other resources) and its information systems. Furthermore, EAs identify the means for collaboration between these different parts, in order to achieve the desired business objectives [19,27]. An ideal EA provides a holistic, enterprise-wide and consistent view of the organization instead of looking at it from the point of view of a single application or system [27,30].

*Corresponding author. Tel.: +358 14 260 3275; Fax: +358 14 260 2544; E-mail: tanja.ylimaki@titu.jyu.fi.

Managing enterprise architectures is, however, a difficult and very complex task. The complexity of systems is typically managed by the division of a problem, i.e. by looking at the system from several viewpoints [7,14,24,28,31]. As such, enterprise architectures, as descriptions of elaborated systems, also comprise a number of different viewpoints (e.g. [38,43]). Some researchers have suggested that, in order to integrate the various viewpoints into a whole, a generic architectural language is needed [15,20,25, 30]. There are also a few frameworks that help navigation in this multifaceted field (see e.g. [2–6,12,16, 18,23,36,40]. The best known, and obviously the most cited, framework is the Zachman framework [38, 43].

In undertaking the writing of this article, we made a considerable effort in searching for scientific research on the Zachman framework. As a result, it seems that there is a lack of scientific studies on the application of the Zachman framework – and analyzing its applicability – in practice (see also [29]). When scientific knowledge is missing, the case study approach is a feasible alternative to reveal unclear or problematic issues in order to instigate more thorough research.

In this article, we present a case study in which the Zachman framework was applied in a situation where an existing, extensive method was adapted to the context of small EA-oriented projects. A small EA-oriented project is defined from the practitioners' point of view: it is a stand-alone project, usually conducted by a single person (e.g. a consultant) aiming at developing a high-level EA specification, a part of the whole EA specification, or providing input to the process of defining such a specification. Moreover, the duration of these projects is typically weeks rather than months. At least in Finland, where the study was conducted, these types of projects seem to be very common, especially in companies taking their first steps in EA development.

The case of applying an existing method in a particular domain is not unique. Information technology (IT) and service providers usually possess a number of methods that are often general yet applied for different purposes. It is acknowledged that tailoring a method is usually needed to meet the actual needs of the development context, but the practitioners lack guidance on how the existing methods, and which parts of them, should be modified [9]. The Zachman Framework is an example of a tool that aids this tailoring process.

Here, we emphasize that we did not have any presumptions about how well and easily the utilization of the Zachman framework would happen in practice. Thus, in contrast to hypothesis-based research, the study was carried out as a combination of practice-oriented application of the Zachman framework and its data-driven analysis.

The problem dealt with in this paper is two-fold. First, there is the practical situation where the case organization encourages the different domain specialists and consultants to apply their extensive in-house method to different kinds of customer projects; in this case small EA-oriented projects. Second, since the Zachman framework seemed to be a suitable aid for such an effort, it awakened our interest for systematically analyzing the process of applying the framework – how easy is it, what kind of "extra" efforts are needed, how practical are the rules of the framework, and so forth. In doing so, we aimed to initiate scientific discussion by eliciting issues that need to be scrutinized further.

We believe that reporting experiences of applying the Zachman framework in practice provides important information on how the existing general frameworks and practical methods can incrementally improve each other. It is not only the practical methods that benefit from mapping with general frameworks, it also works in the other direction; the validity of general frameworks can be improved by examining examples taken from practice.

The paper proceeds as follows. In the next section we give a brief introduction to the Zachman framework. Following this, we describe the case study. Finally, we discuss the implications of the study and present some suggestions for further research.

Table 1
The Zachman Framework for Enterprise Architecture (based on [38,43])

	DATA (What?)	FUNCTION (How?)	NETWORK (Where?)	PEOPLE (Who?)	TIME (When?)	MOTIVATION (Why?)
SCOPE (Contextual) *Planner*	List of things important to the business	List of processes the business performs	List of locations in which the business operates	List of organizations important to the business	List of events significant to the business	List of business goals/strategies
BUSINESS MODEL (Conceptual) *Business Owner*	E.g. Semantic Model	E.g. Business Process Model	E.g. Business Logistics System	E.g. Work Flow Model	E.g. Master Schedule	E.g. Business Plan
SYSTEM MODEL (Logical) *Architect*	E.g. Logical Data Model	E.g. Application Architecture	E.g. Distributed System Architecture	E.g. Human Interface Architecture	E.g. Processing Structure	E.g. Business Rule Model
TECHNOLOGY MODEL (Physical) *Designer*	E.g. Physical Data Model	E.g. System Design	E.g. Technology Architecture	E.g. Presentation Architecture	E.g. Control Structure	E.g. Rule Design
DETAILED REPRESENTA-TIONS (Out of context) *Sub-Contractor*	E.g. Data Definition	E.g. Program	E.g. Network Architecture	E.g. Security Architecture	E.g. Timing Definition	E.g. Rule Specification
FUNCTIONING ENTERPRISE	E.g. Data	E.g. Function	E.g. Network	E.g. Organization	E.g. Schedule	E.g. Strategy

2. The Zachman framework

Zachman introduced his framework for Information Systems Architecture in 1987 and it is usually referred to as the Zachman framework ([43]; extended by [38]). The starting point for the framework was the fact that information systems were becoming ever more complex. There was a threat that without any integration tools, information systems would disintegrate rather than integrate business functions. Zachman draws an analogy between classical architecture (construction of buildings) and information systems architecture. The two ideas behind the framework are, as Zachman puts them, that:

– There is a set of architectural representations produced over the process of building a complex engineering product, representing the different perspectives of the different participants.
– The same product can be described, for different purposes, in different ways, resulting in different types of descriptions [43, p. 283].

Based on these two ideas, the Zachman framework combines two dimensions: the *perspectives* and the *types of descriptions* (see Table 1). The perspectives (or views) form the rows and the types of descriptions (or foci) form the columns of the framework.

As described in Table 1, the different perspectives of the two-dimensional matrix are Scope, Business Model, System Model, Technology Model, Detailed Representations, and Functioning Enterprise. These represent different interest groups, which are the planners, owners, designers, builders, and sub-contractors, respectively. The types of descriptions used in the framework are Data, Function, Network, People, Time and Motivation. These represent the interrogatives What, How, Where, Who, When and Why, respectively.

The significant feature of the framework is that each of the elements on either dimension of the matrix is explicitly differentiable. So, each cell of the framework has its own way of “seeing” the object architecture.

The Zachman framework provides both a tool to organize enterprise architecture documentation and a context for understanding the relationships between and among separate sets of architectures [43]. A further important feature is that the framework includes both contextual and enterprise level descriptions, and the descriptions needed at the design and implementation level of an information system. The framework is defined with total independence from any particular tool or methodology; it does not instruct how to build an appropriate EA specification. As such, it provides a general framework onto which an EA tool or methodology can be mapped and appropriate modeling techniques can be chosen to describe the cell contents. This is natural because the process of developing an EA is, to a great extent, dependent on the context. The purposes of EA specifications vary markedly, as do the resources available for building EAs.

Although there is a lack of scientific empirical studies on the utilization of the Zachman framework in practice, some research has been carried out in an attempt to either describe or analyze the framework [33, 34] or use it as a method assessment tool [8,10,40]. In many cases, the Zachman framework has been used as a "baseline" for developing new, modified or simplified frameworks [1,13,26,32]. Some new methods utilizing either a part of the Zachman framework (e.g. [17]) or the whole framework (e.g. [35]) have also been developed. In general, it seems that the Zachman framework has reached the status of a de-facto standard. Nonetheless, scientific reports providing guidance or analyzed examples of *applying and utilizing* the Zachman framework *in practical cases* are hard to find.

As a prerequisite for understanding and applying the Zachman framework, some basic rules of the framework need to be presented. These are described by Sowa and Zachman [38] and they include, for example, the following:

- The columns have no order. All columns are equally important, for all are abstractions of the same enterprise. The order of the rows is fixed.
- Do not add rows or columns to the framework. Zachman claims that the six primitive interrogatives (what, how, where, who, when, why) constitute the total knowledge base for the subject (or object) you are describing. They are also comprehensive, with additional interrogatives adding no new information.
- Each cell in a row should include an architectural primitive, a single (graphical) model describing the enterprise from a certain point of view (the six types of descriptions). Each cell is unique.
- One can choose not to produce deliverables for every cell. In that case, one makes assumptions about them. All columns, rows and cells are always present in the framework, though some cells can be left implicit.
- The composite or integration of all cell models in one row constitutes a complete model from the perspective of that row. Each cell is, as a minimum, related to every other cell in the same row. Within a column there is a relation of dependence between any one cell and both the cell above and below it. Diagonal relationships should not be created.

Zachman suggests that EA is the "set of primitive, descriptive and graphically presented artifacts that constitute the knowledge infrastructure of the enterprise" [45]. This total set of models may be created in a non-standardized manner and it may be more or less complete, as well as being defined with an inaccurate level of detail. No matter how complete or incomplete these models might be, they must be descriptive of the enterprise – not just descriptive of an implementation within the enterprise – to qualify as an EA.

In contrast, the application development work products are created as inputs or outputs for the application development process, and thus, for implementation purposes [45]. They constitute the actual work

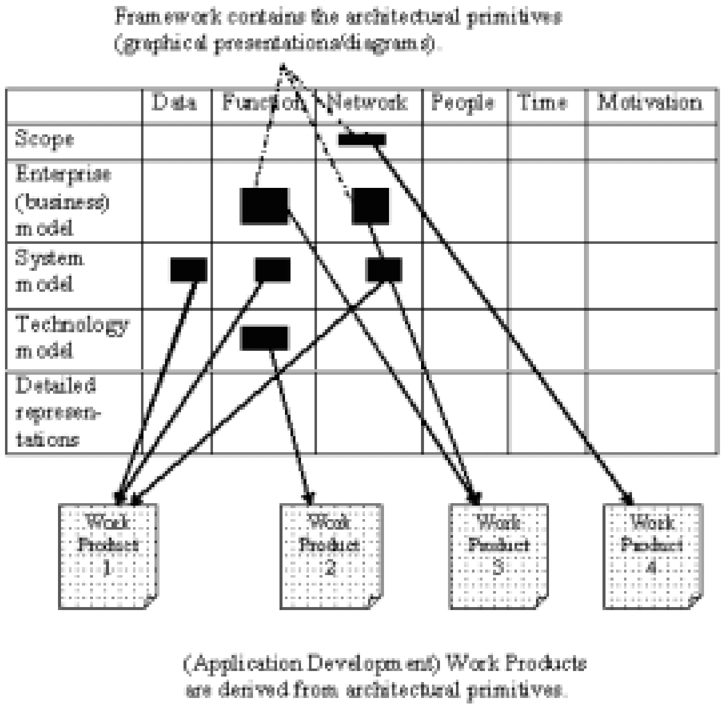

Fig. 1. The Zachman framework should be comprised of architectural primitives from whom the application development work products can be derived.

products for building information systems. EA is the set of primitive artifacts (see Fig. 1) from which application development work products could be derived [46].

In the following section we will describe the case study in which we applied the Zachman framework to small EA-oriented projects in conjunction with an extensive, existing method, namely the IBM Global Services Method*. This context made the utilization of the Zachman framework challenging because it seemed to require a dialogue between the framework and the in-house method.

3. The case study

In this section we describe the case study in which the Zachman framework was applied. First, we give a short description of the case organization. Second, we represent the method that was mapped against the Zachman framework. Third, we describe the research process and, fourth, we represent the resulting lightweight framework for small EA-oriented projects, its evaluation and use cases.

3.1. The case environment

The target organization of the case study is one department at IBM Finland. IBM supports its main business areas by providing a number of methods to cover the whole customer project from the first contact with the customer to the lessons learned at the end of the project. There are methods that provide guidance for such things as customer focused selling, project management, and the design and implementation of business solutions. One of the newest methods is the IBM Global Services Method*.

This method is intended to be used in various types of customer projects (e.g. business strategy or IT architecture projects) providing standardization for delivering overall business solutions.

The practical target of our study was to provide the practitioners with some help in applying the IBM Global Services Method* to small EA-oriented projects conducted in Finland, including both IT strategy aspects and EA aspects. High-level EA is one outcome of the project. The basic, high-level steps of these projects are the following: (1) initiate the project (a project management issue), (2) understand the business processes, (3) understand the current IT environment, (4) design the target architecture (application and technology architecture), (5) design the organization (an optional phase), (6) develop a transition plan, and (7) end the project (a project management issue).

Because the IBM Global Services Method* is an extensive method, providing almost everything that is needed in large scale projects, the pressing question is how to distill the pieces of the method that are necessary for a small EA-oriented project.

The structure of the method is described in the next section.

*3.2. The IBM Global Services Method**

The IBM Global Services Method* (later IGSM) is used as a design and implementation method for various business solutions. It provides the project context – what is sold and managed, what the team will do and what they will deliver. Even though the method is not publicly available in any form other than through a contracted IBM engagement, it has been described briefly by, for example, Singer [37] and Galic et al. [11].

The method consists of three main components [22]: work product descriptions (WPD), engagement models (EM) and standard role definitions (see Fig. 2). In short, the WPDs advise 'what' to do, EMs advise 'how' to conduct a certain type of project (when things need to be done) and standard role definitions advise 'who' does the work. In addition to these components, there are many technique papers (TP) that provide problem specific insight for the development of a work product in different circumstances.

As the method is constantly evolving, changes may occur in the work product descriptions as new ones are defined and some of them are superseded or withdrawn. Method release 4.1.1, for example, includes about 400 WPDs [22]. These describe a particular type of work product (WP) and give guidance on how to create the actual work product, what notation can or should be used, and the recommended structure of the work product itself. A work product is meant to be a tangible, reusable artifact, produced as a result of one or more tasks performed in a project. Moreover, to help manage such a wide range of information, WPDs are categorized into several domains and sub-domains (see Fig. 3).

The selection of the actual WPs to be produced in each project is eased by work product dependency diagrams (WPDD). WPDDs show the key work product dependency relationships that need to be considered. These diagrams provide a point of reference for making decisions about which WPs the team needs to produce in a specific customer project.

WPDs are employed by engagement models that provide guidance on the way that projects (or engagements in the IBM context) should be conducted. Due to the continuous development of the method, the content and number of EMs will vary over the course of time. Currently, there are about 100 EMs available for different types of projects [22]. The EMs are classified into engagement families. These families represent the types of projects that are carried out, or the actual product or service sold (see examples in Table 2).

Each engagement model provides guidance for the phases, activities, and tasks required according to the work breakdown structure (WBS), the work products that are needed (input) and produced (output),

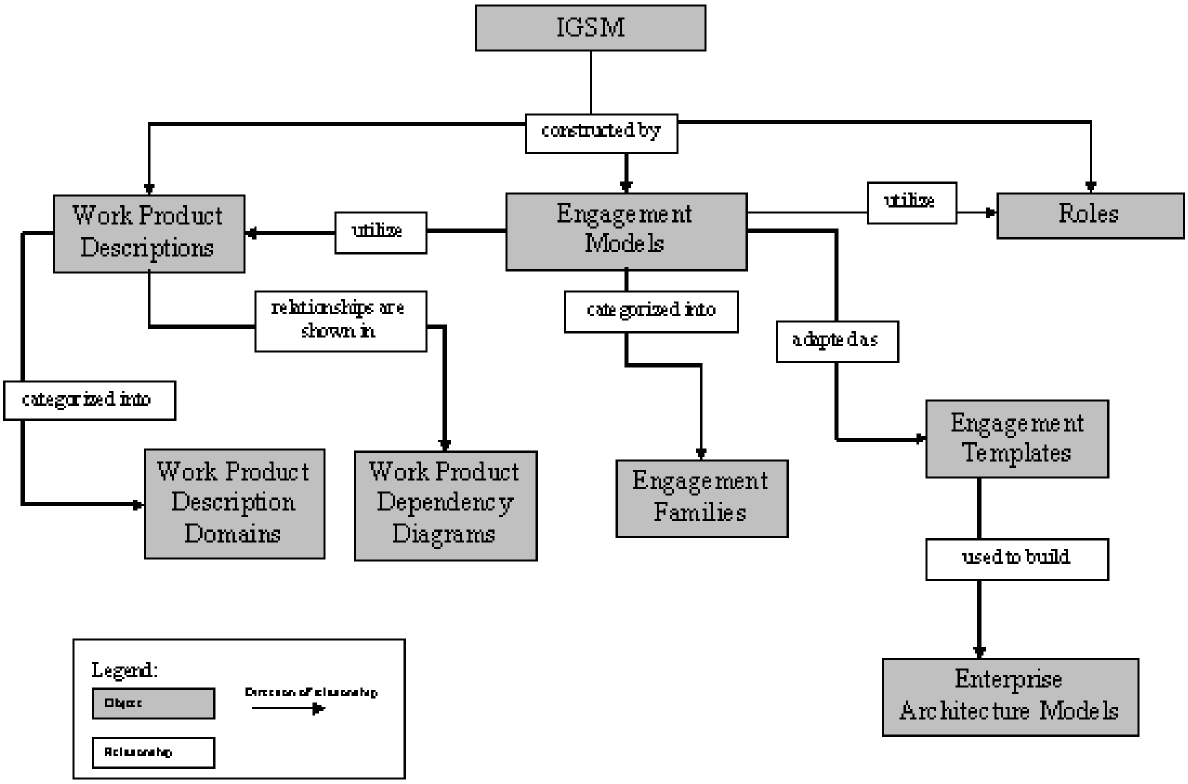

Fig. 2. IBM Global Services Method* is constructed by work product descriptions, engagement models and roles.

and any applicable techniques that should be used for one or more of the tasks. Furthermore, for each task in an engagement model, the roles describing the skills required to perform and assist in the task are defined. The definitions of the standard roles are used to ensure a greater level of consistency and standardization. An extract of an engagement model is illustrated in Fig. 4.

According to the IGSM, a method adoption workshop (MAW) is arranged in the beginning of the project to agree on the overall approach that will be taken in the client engagement. In the MAW, the engagement models chosen for the project are discussed and the relevant parts are selected and possibly modified (phases, activities or tasks can be omitted, combined or planned to be carried out with minor efforts). This adoption work requires both method and project experience and it is supported, for example, by the WPDDs. As a result of the MAW, an engagement template is generated providing a WBS that will be followed in the project to develop (enterprise architecture) models and descriptions in the form of WPs.

3.3. Research process

The decision to utilize the Zachman framework was made in a meeting between the researchers and the company representatives. In Fig. 5, the successive steps of the case study are depicted. These steps are as follows:

- *Selection of WPs*: The WPs to be mapped against the Zachman framework were chosen. As the method includes two EA related engagement models, they were a natural starting point for WP selection.

Table 2
Some examples of engagement families of the IGSM

Engagement families
1. Custom Application Development
2. Customer Relationship Management
3. e-business
4. Enterprise Architecture Consulting
5. Knowledge and Content Management Services
6. Mobile and Wireless Consulting
7. Project Management
8. Solution Consulting and Integration

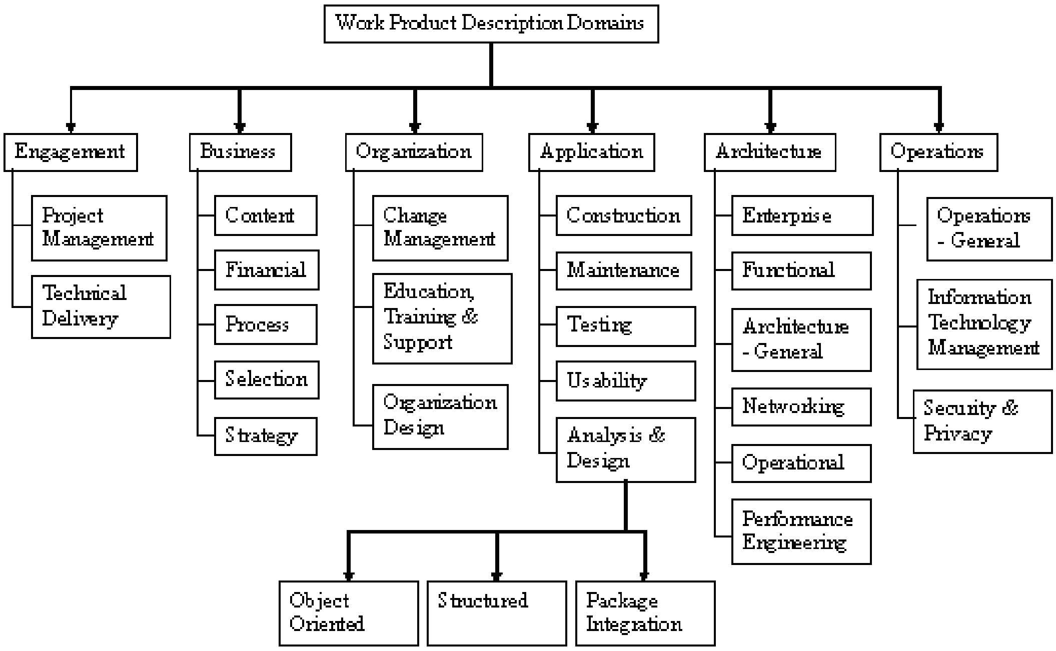

Fig. 3. Work product descriptions are categorized into 6 high level domains and further into several sub-domains.

- *Definition of mapping rules*: The rules of the Zachman framework were studied and some modifications and exceptions were made to be able to do the mapping between the WPs of the method and the Zachman framework.
- *Mapping the WPs against the Zachman framework*: The selected WPs were mapped against the Zachman framework cells according to the rules defined in the previous step. As a result, a candidate lightweight framework was accomplished.
- *Evaluation of the lightweight framework*: Evaluation was conducted in two stages: (1) The candidate lightweight framework (the initial mapping) was reviewed and suggestions for modifications were made in two successive workshops in which the project group, consisting of four company representatives, participated. (2) A further evaluation of the reviewed lightweight framework was done by a single method expert (from IBM) as a “table test”. Because no suitable customer project was, unfortunately, at the time available to test the lightweight framework in practice, we had to settle for this procedure.

- Engagement Model: IT Architecture
  - Phase 1 …
  - Phase 2 …
  - Phase 3: Understand Business and IT Context
    - Activity 1 …
    - Activity 2 …
    - Activity 3: Initiate High Level IT Architecture Definition
      - Task 1: Define IT Strategic Vision
        - Role (performs): IT Strategy Consultant
        - Role (assists): Business Analyst
        - Role (assists): Infrastructure Consultant
        - Work Product (input): Business Direction
        - Work Product (input): Business Drivers
        - Work Product (input): Capability Model Definition
        - Work Product (input): Technology Scan
        - Work Product (output): Architecture Overview Diagram
        - Work Product (output): Critical Issues, Opportunities and Recommendations
      - Task 2 …
    - Activity 4 …
  - Phase 4 …
  - Phase 5 …

Fig. 4. An extract of the work breakdown structure of the IT architecture engagement model describes the hierarchy of phases, activities, tasks, roles and input/output work products [22].

3.4. Selection of work products

The IBM Global Services Method* includes an engagement family for enterprise architecture consulting and within the family there are two engagement models: one is used to define a high level enterprise-wide IT architecture and the other is used to develop a description of how a business should be structured in order to implement and support the business strategy of the organization [22]. The first engagement model can be adapted, for instance, for a short stand-alone engagement (approx. 6 weeks) when a client requires an initial enterprise level IT architecture to be defined, as the first step in developing and implementing an overall EA, or as part of an overall EA engagement. The latter engagement model is also applicable as the initial phase in the development of a complete EA, or in situations where only a business architecture is required.

Since the two engagement models in the engagement family for enterprise architecture consulting comprise the overall EA within the IGSM context, they were a natural starting point for selecting the WPs. Additionally, during the earlier phases of the research project, a few other WPs were considered by the practitioners to be important even though they were not included in the two EA-related engagement models. Some of the additional WPs were also recommended by the IBM Architecture Description Standard [21]. These WPs were also included in the list of WPs to be mapped.

The engagement models also include project management tasks and work products. Our focus was, however, on the WPs that comprise the content of the EA (the EA descriptions). Hence, we excluded the project management work products from the list to be mapped. The resulting list of 52 work products, with brief descriptions, is presented in Table 3.

3.5. Mapping rules

We acknowledge the strong role of the Zachman framework as an EA tool. In a similar way to others (see e.g. [24]), we regard the framework as a useful *visual* tool for communication or organization in

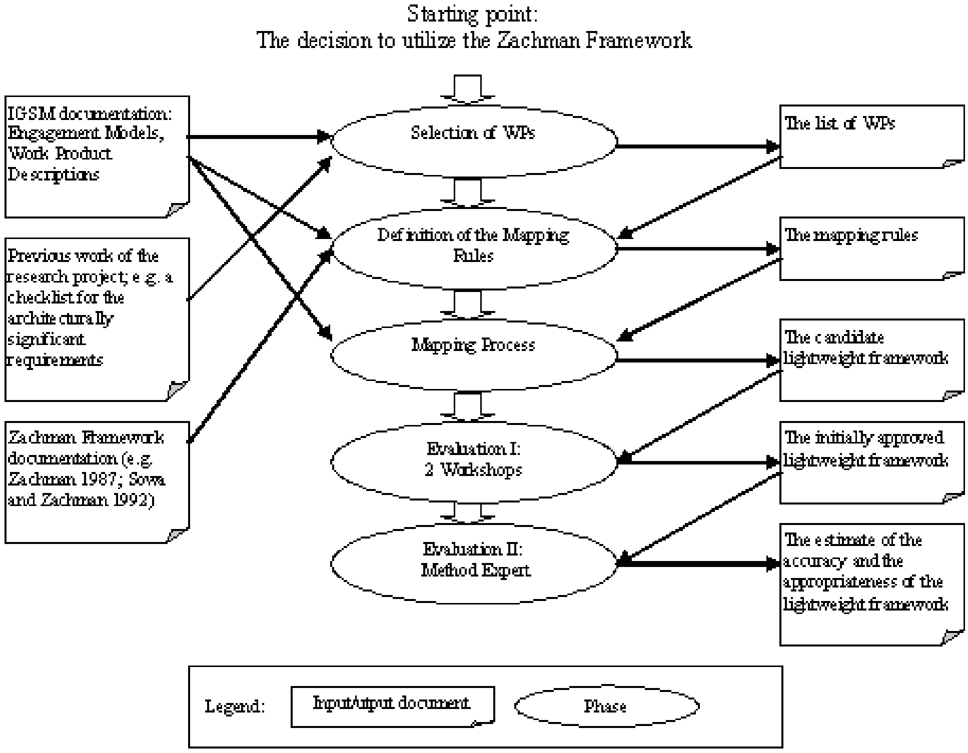

Fig. 5. Steps of the case study conducted in one unit of IBM Finland.

EA development. In particular, the framework provides a big picture of what it is included in enterprise architecture and urges the EA team to take all the framework cells into consideration in large EA projects [44]. However, during our three-year research project, which this case study is part of, we came to the conclusion that the framework has at least two problems: (1) it is rather complex to be utilized without modifications, and (2) it does not sufficiently analyze the interconnections between the cells (see also [29,39]). Additionally, as mentioned before, the scientific reports or publicly available practical guidance on utilizing the Zachman framework are rare.

In our case, we needed to take the elements (work products) of the extensive in-house method as a fixed starting point, even though it was obvious that many of the work products comprised components from more than one cell of the framework, and thus, as Zachman implies, they should not be mapped onto the framework. Instead, the primitive models within the work products should be determined before placing them in the appropriate cells in the framework. In an ideal situation, with plenty of time and other resources, the mapping would include only primitive models; nonetheless, in our case – which we assume is a typical case in practice – we had limited resources both in the sense of time and workload and, subsequently, following the strict demand for the atomicity of the cells seemed unfeasible.

Nevertheless, in the Zachman framework we had a good benchmarking tool rather than a competing methodology that defines the rules to which the EA artifacts should conform. Therefore, we considered the Zachman framework as a means to check the coverage of EA-related work products. From our point of view, the way in which an EA specification would be built is not relevant.

Due to all of these aforementioned facts, it seemed obvious that using the original mapping rules would have made the task too tough to complete within the confines of reasonable effort. Hence, we made a

Table 3
The set of 52 work products that are mapped against the Zachman framework in an alphabetical order

Work Product	Description
1. Application Function Model	Identifies and defines the major groups of business function that are required in order for the enterprise to meet its business objectives.
2. Architectural Decisions	Documents important decisions about any aspect of the architecture.
3. Architecture Management Framework	Defines the processes, roles and responsibilities required to manage and implement an enterprise-wide architecture.
4. Architecture Overview Diagram	A schematic diagram that represents the governing ideas and candidate building blocks of an enterprise architecture (or information system).
5. Business Direction	Includes a high-level understanding of what the enterprise is and aspires to become, and of how it intends to make the desired transition.
6. Business Drivers	Identifies the top business issues, challenges or priorities for the enterprise.
7. Business Environment	Illustrates a business entity in context with its external environment.
8. Business Event List	Documents the events that are the initial stimuli causing the client's business to act.
9. Business Roles and Locations	Documents the roles and locations that are pertinent to the enterprise's business.
10. Business Structure	Is a model that is used to identify e.g. the sub processes or deployment architecture.
11. Capability Description	Contains the information describing a capability and its attributes.
12. Capability Enablers	Documents the physical things that must be present in the future design to enable the capabilities identified to deliver on the value proposition.
13. Capability Model	Provides a definition and description of each selected business or IT capability that supports the value propositions.
14. Capability Model Definition	Depicts an integrated framework of a firm's capabilities.
15. Capability Scenario	Clarifies how the business will work with the new set of capabilities defined in the capability model.
16. Change Cases	Documents future changes e.g. to the system capabilities and properties and the way the system is used.
17. Communications Plan	Details the actions to be taken to fulfill the communication role identified in the Transition Strategy.
18. Critical Issues, Opportunities and Recommendations	An aggregation of highly important problems and/or opportunities for which effective solutions must be created or provided, and recommendations to address these issues and opportunities.
19. Current IT Environment	Documents the installed applications, data, computer and network infrastructure, and the current status of all the non-technical, management aspects.
20. Current Organization Description	An inventory of organizational information on the elements of an organization structure, behaviors and enablers for the in-scope organizational units.
21. Data and Function Assess and Placement	Provides guidelines that must be applied when distributing data and function across the IT infrastructure implemented within the enterprise.
22. Data Stores	Identifies, describes and groups business data in terms of its characteristics.
23. Decision Model	An analytical hierarchy of the criteria used in evaluating the alternative solutions to the client business requirements.
24. Enterprise Information Model	Represents the strategic information requirements of the enterprise.
25. Enterprise Technology Framework	A repository for all information about the IT capabilities and enablers required to implement the desired business objectives and capabilities.
26. Executive Briefing Package	An executive-level presentation of business concepts and information designed to educate the client executives on trends, practices, and facts that will drive decision making.
27. Future Business Environment	Identifies the critical trends and relevant environmental forces that are shaping the issues facing the organization and influencing the outcome of the organization's strategic decisions.
28. Future Organization Design	Details the new structure, performance enablers, and behaviors required to achieve the business vision.
29. Future Organization Scope and Requirements	Describes and gains agreement to e.g. the geographies, divisions, and levels to be addressed in redesign, and the elements of the organization to be addressed in redesign.
30. Industry Environment Analysis	Describes an understanding of the industry environment in which the client organization is operating.
31. Infrastructure Gap Analysis	Identifies the gaps between the existing technology, applications, processes, skills, and organization and those required to implement the new information system.

Table 3, continued

Work Product	Description
32. IT management Requirements	Documents all the requirements needed to deliver the in scope IT management services.
33. Job Roles – Responsibilities and Competencies	Describes all of the elements that constitute an employee's role in the future organization.
34. Knowledge Gap	Documents the gaps between the existing knowledge management environment and its desired practices.
35. Network Requirements	Contains the description of the specific network requirements needed to support the desired business, user or application objectives.
36. Non-Functional Requirements	Documents those aspects of the system that are not directly affecting the functionality of the system as seen by the users, but can have a profound effect on how the system is accepted by the users.
37. Operational Model	A representation of a network of computer systems, their associated peripherals and the systems software, middleware, and application software that they run.
38. Principles, Policies and Guidelines	Defines the underlying gen ules and guidelines that an organization will use to utilize and deploy all business and IT resources and assets.
39. Process Data Usage	A bi-directional mapping between the enterprise processes and enterprise information.
40. Process Definition	Identifies and describes the current and/or future processes, sub-processes, activities and steps.
41. Process Enablers	Documents the prioritized business requirements for the future organization and technology, risks and assumptions that have been made.
42. Process Gap Assessment	Documents the gaps between the current processes and the future processes.
43. Process Identification	Identifies the processes that are necessary to execute the organization's business e.g. at the enterprise level.
44. Reference Architecture Fit-Gap Analysis	Documents the Reference Architecture to be used as the basis for the current project's architecture.
45. Security and Privacy Requirements	Documents the process and technology needs specific to the client's bus iness and IT strategies.
46. Stakeholder/Participation Management Plan	Identifies and sets out approaches for preparing and involving stakeholders during a project implementation.
47. Standards	Lists and details pre-determined standards for a defined scope of the business and the information technology environment.
48. Strategic Direction	Includes a high-level understanding of what the enterprise is and aspires to become, and of how it intends to make the desired transition (e.g. vision, mission, goals).
49. Technology Scan	Offers benchmarking or comparative baseline view to identify e.g. new information technologies that offer value for the business.
50. Transition Initiatives	Pulls together various business and IT initiatives, projects, and transition issues defined in previous engagement activities.
51. Transition Management Strategy	Summarizes the needed changes and defines the overall approach that an organization will take to implement its new capabilities.
52. User Groups	Lists categories of users.

decision to modify the rules. It should be made clear that this decision was not made for convenience but, rather, from necessity. Nonetheless, we wish to underline that we did not change the boundaries of the Zachman framework as a whole.

In the following list we introduce the exceptions made to the rules of the Zachman framework and some additional rules we used to map a WP against the appropriate framework cell(s):

- A work product can be placed into more than one of the cells, i.e. a work product can even be produced at more than one level of abstraction (rows in the framework). A work product can also cover different points of view (columns in the framework).
- Work products are attached to a cell on the basis of the information given in the work product descriptions (the method users' experience is also used to confirm that the work product is in the right cell).
- Some work products may relate to the whole framework, e.g. Architecture Management Framework.

Such work products are not included in any of the cells, but are mentioned at the bottom of the matrix.

– Some work products containing requirements, constraints or some kind of rationale or motivation are attached to the proper row(s) in the Motivation column, even though they can include information related to other columns as well. Examples of these kind of work products are:

  * Process Enablers,
  * Business Drivers, and
  * Reference Architecture Fit-Gap Analysis.

3.6. *The lightweight framework for small EA-oriented projects*

In line with the research process, the lightweight framework was accomplished by mapping the set of 52 EA related work products to the Zachman framework cells according to the rules discussed in the section above, and this initial mapping was reviewed in two workshops in which the company representatives participated. As a result, we got a lightweight framework that describes the content of small EA-oriented projects in the context of the case organization (Table 4). We call it a 'lightweight' framework because instead of wading through the massive method content (including a total of approximately 400 WPDs) a practitioner can utilize it as an alternative and simple view of the method, presenting a selection of WPs related to EA. A further selection of the most salient WPs – usually less than ten – for a specific small EA-oriented project can be done with the help of this set.

In addition to the rules and the cell examples of the Zachman framework, the work product description (WPD) documents were studied to determine the row(s) and column(s) in which the work products (WP) should be placed. Because the method content is confidential, the extent of details we are able to describe about the mapping process is limited. Some rough examples can be given, though: (1) If a WPD stipulated that a WP can either be a high-level description or a more detailed description depending on the project's needs, the WP was mapped to the appropriate rows of the framework (e.g. the Enterprise Information Model was mapped to the rows 'Scope' and 'Business Model' in the 'Data' column). (2) If a WPD stipulated that a WP describes several issues, such as data, function and network, the WP was mapped to the respective columns of the framework (e.g. the Architecture Overview Diagram was mapped to the columns 1–3 in the row 'Business Model'). Compared to Zachman's ideas, the only thing we did not do was to elucidate the atomic artifacts related to the WPs due to the significant difference between the granularity of the WPs and Zachman's demand for atomic artifacts (as well as the limited resources, both in the sense of time and workload).

The two bottom rows in the Zachman framework, namely 'Detailed Representations' and 'Functioning Enterprise', are excluded from Table 4 because they represent the implementation view and no work products were included in those rows.

It should be noted that while IGSM and, therefore, the engagement models are constantly evolving, and new versions of the method are introduced, some of the work products used in the mapping may already have been superseded, modified, renamed or withdrawn from the method.

3.7. *Evaluation of the lightweight framework*

In order to be utilized in customer projects by other IBM representatives, besides those who participated in this study and contributed to the mapping, the accuracy and the appropriateness of the lightweight framework, the mapping, should be evaluated and validated in real customer cases. Unfortunately, during

Table 4
The lightweight framework for small EA-oriented projects: A set of work products are mapped against the Zachman framework

	DATA (What?)	FUNCTION (How?)	NETWORK (Where?)	PEOPLE (Who?)	TIME (When?)	MOTIVATION (Why?)
SCOPE (Contextual) *Planner*	– Capability Enablers – Change Cases – Enterprise Information Model	– Capability Enablers – Change Cases – Process Identification	– Capability Enablers – Change Cases – Business Roles and Locations	– Capability Enablers – Change Cases – Business Roles and Locations – Current Organization Description – Future Organization Scope and Requirements – Stakeholder/Participation Management Plan – Communications Plan	– Capability Enablers	– Capability Enablers – Business Direction – Business Drivers – Strategic Directions – Executive Briefing Package – Reference Architecture Fit-Gap Analysis – Critical Issues Opportunities and Recommendations – Business Environment
BUSINESS MODEL (Conceptual) *Business Owner*	– Enterprise Information Model – Knowledge Gap – Architecture Overview Diagram – IT Management Requirements – Process/Data Usage – Data Stores	– Process Definition – Business Structure – Architecture Overview Diagram – IT Management Requirements – Process/Data Usage – Process Gap Assessment	– Business Structure – Architecture Overview Diagram	– Future Organization Design – Job Roles – Responsibilities and Competencies – Architecture Overview Diagram – IT Management Requirements	– Business Event List	– Principles – Policies and Guidelines – Capability Description – Capability Model Definition – Capability Model , – Capability Scenario – IT Management Requirements – Future Business Environment – Industry Environment Analysis – Decision Model – Process Enablers – Standards
INFORMATION SYSTEM MODEL (Logical) *Architect*	– Data Stores – Process/Data Usage – Data and Function Access and Placement – Enterprise Technology Framework	– Application Function Model – Data and Function Access and Placement – Enterprise Technology Framework	– Network Requirements – Data and Function Access and Placement – Enterprise Technology Framework	– User Groups		– Standards – Transition Initiatives
TECHNOLOGY MODEL (Physical) *Designer*	– Enterprise Technology Framework – Technology Scan – Non-Functional Requirements – Current IT Environment – Infrastructure Gap Analysis	– Enterprise Technology Framework – Technology Scan – Non-Functional Requirements – Current IT Environment – Infrastructure Gap Analysis	– Enterprise Technology Framework – Technology Scan – Non-Functional Requirements – Current IT Environment – Infrastructure Gap Analysis – Operational Model	– Technology Scan – Non-Functional Requirements – Current IT Environment – Infrastructure Gap Analysis	– Technology Scan – Non-Functional Requirements	– Standards – Technology Scan – Non-Functional Requirements – Current IT Environment – Infrastructure Gap Analysis – Transition Management Strategy

WPs that relate to the whole matrix: Architecture Management Framework, Architectural Decisions, Security and Privacy Requirements

Table 5
Evaluation data was classified into five categories representing the accuracy of the mapping

	Agree with the mapping	Agree but would include/exclude column(s)	Agree but would include/exclude row(s)	Would include/ exclude column(s) and row(s)	Would not map at all	Total
Number of WPs	21	4	12	2	13	52
Percentage	40%	8%	23%	4%	25%	100%
Cumulative number of WPs	21	25	37	39	52	
Cumulative percentage	40%	48%	71%	75%	100%	
Number of the mappable WPs	21	4	12	2	–	39
Percentage	54%		46%			100%

the case study we were not able to make any evaluation in real customer project context. As previously mentioned, the initial mapping was reviewed and evaluated in two workshops in which the project group, consisting of four company representatives, participated. In addition, we were able to have an IBM Global Services Method expert (from IBM) look at the reviewed mapping and evaluate its accuracy. In this section, we describe the results of the evaluation conducted by the method expert.

The evaluation data received from the evaluator was classified in the following five categories:

1. *Agree with the mapping*: The evaluator agreed with the mapping of a WP as it was.
2. *Agree but would include/exclude column(s)*: The mapping of a WP was appropriate for the row(s), but it should also be included under the column(s) NN, or it should not be included under the column(s) NN.
3. *Agree but would include/exclude row(s)*: The mapping of a WP was appropriate for the column(s), but it should also be mapped to the row(s) NN, or it should be mapped to the row(s) NN instead.
4. *Would include/exclude column(s) and row(s)*: There should be changes in both the columns and rows of the mapping of a WP.
5. *Would not map at all*: The evaluator suggested that a WP should not be mapped at all. In this case, the WP was either outdated or irrelevant with respect to the current version of the IGSM and/or EA, or it otherwise did not relate well to the Zachman framework.

To clarify the categorization we give an example of categories 2–5. In category 2, the mapping of 'IT Management Requirements' (see Tables 3 and 4) is otherwise relevant, but it could be included under the 'Network' column also. Respectively, in category 3, 'Architecture Overview Diagram' could represent the artifacts in the 'Scope' row. In category 4, the work product, 'Application Function Model', can also be mapped to the 'Data' column, and to the respective columns in the 'Business Model' row. In category 5, there are WPs that are not recognized as EA WPs in the current method version (e.g. 'Operational Model'), or they are otherwise not related to the Zachman Framework (e.g. 'Executive Briefing Package'). The results of the evaluation are summarized in Table 5.

As Table 5 shows, the evaluator agreed that 75% of the mapped WPs have their place in the Zachman framework, but suggested that 25% of them should not be mapped at all. In the latter set, there were work products that were considered either to be irrelevant from the EA point of view, were outdated and withdrawn from the method (IGSM), or they otherwise did not relate well to the Zachman framework. In other words, this means that the majority of the WPs relevant to both the IGSM and EA can be mapped onto the Zachman framework. On the other hand, out of the 75% of the WPs – the mappable WPs – 54% were agreed upon 'as-is' by the evaluator and 46% of the mappings required changes in the columns and/or rows. Thus, it can be concluded that the Zachman framework helped relatively well to find the relevant content and position in half of the cases, whereas in the other half it was not easy to decide where to place a work product, not to mention putting it into a single cell.

We acknowledge that the further evaluation done by a single expert may have some subjective emphasis, and more evaluators will be needed to validate the correctness of the mapping. The evaluation conducted by the method expert, together with the earlier reviews carried out by the company representatives, proved the mapping to be accurate enough for our purposes, i.e. for analyzing the applicability of the Zachman framework. From a practical point of view, even though the mapping as-is would not be absolutely accurate, it is informative enough to be utilized in small EA-oriented projects employing, for example, less than three people. In these kinds of projects, the content (the WPs included in the framework) is more essential than the actual correctness or the completeness of the mapping because the structure of the EA specification is relatively easy to manage. The level (the row) or the views (the columns) into which a WP is mapped is usually dependent on the context it is applied to, that is to say, the customer project. Even the first version of the lightweight framework presented in this paper provides a means, for example, to enable more fluent communication among different stakeholders or to guide the WP selection in the beginning of the project. In larger EA efforts, the validity of the mapping has, instead, a more crucial role. As the number of the team members increases, and work is distributed and delegated to several persons, it is important that each person interprets the mapping in a similar way. From the point of view of research, however, the most relevant results come from the experiences gathered from the mapping process.

3.8. Usefulness of the lightweight framework

The lightweight framework presented in this paper can be used in several ways. We suggest that, in the case organization, it can be used as:

- A customer discussion aid, helping to clarify the scope of a project (e.g. which rows and columns are the concern of the potential project)
- A method adoption tool, helping to decide which architectural work products are needed to be produced during a project, or as a baseline for defining minimum work product sets for different types of enterprise architecture oriented projects
- A checklist helping to verify that all the work products needed have been done
- An engagement assessment tool, to gather information about the success of a project.

Customer discussion aid: The lightweight framework can be used to help the customer to understand what perspectives the enterprise architecture consists of, what the focus of a prospective project is, and how it relates to the entire enterprise architecture (i.e. the entire matrix). For example, one project may focus on rows one and two, while another project may focus on column one or column two.

Regardless of the focus, every aspect of the enterprise architecture should be taken into consideration as the adjacent rows set bounds to each other and the cells in the same row are potentially dependent upon each other. If all of these aspects cannot be fully defined within the scope of a project, assumptions (or good guesses) have to be made about them.

Method Adoption tool: The lightweight framework can be used to help the selection of the actual work products to be produced during the EA-oriented engagement. Depending on the focus and size of the project, relevant work products can be chosen from the appropriate rows and/or columns in the framework. As the framework does not imply the dependencies between work products, those dependencies must be checked with other appropriate tools, such as the work product dependency diagrams.

Checklist: During the later stages of the project, the modified lightweight framework may be used as a checklist to ensure that all the relevant work products have properly been produced. Different symbols can be used to describe whether a work product is finished, under development or not even started.

Engagement assessment tool: Information about the usefulness of the lightweight framework for the project success – the lessons learned – can be gathered and evaluated at the end of the project. For example, the following questions can be answered:

- How well did we succeed in selecting the appropriate work products for the project? Was there a need to add or remove some work products during the project? Which work products? Why? Was the scope of the project properly defined?
- How did the project type relate to the framework? Which cells and/or work products were covered by the project? Were there similarities in the coverage compared to other projects of this type? Is it possible to define a minimum set of work products for different project types?
- Ideas for future projects: for example, which cells remained to be covered in further engagements?

The ideas for use cases presented above can be applied to other similar cases where an existing method is mapped against the Zachman framework.

4. Discussion

In this section, we discuss the findings of the case study and the implications for both researchers and practitioners.

As a general conclusion, the Zachman framework proved a useful mapping tool, despite the fact that the mapping rules had to be modified. As the evaluation of the developed lightweight framework shows, the mapping was quite accurate. The mapping was also relatively easy to do, although the method under consideration was very extensive. We believe that the main reason for this was just the elaboration of the mapping rules mentioned above. Without any modification to the rules, the task would obviously have been too time-consuming.

We are very conscious of the risks of interpreting the Zachman framework in the unorthodox way in which we did, but there were justifiable grounds for doing so. First, it is hard to find good examples of the rules of the framework being applied in practice. Thus, it seems to us that the Zachman framework has often been utilized as a general EA framework rather than an explicit methodology – the matrix itself is known better than the rules guiding the use of it. If this is the case, it may imply that the mapping rules do not have as prominent a role in practice as one would assume. Second, while the framework provides an architecture for architectures, this does not necessarily mean that in order to have a similar building one always has to use similar elements. In our case, the Zachman framework provided the model for the building, but the material (elements) for the building was determined by the case organization (the elements of the in-house method). To utilize the Zachman framework in the best way, we attempted to facilitate *interaction* between the Zachman framework and the in-house method of the case organization. In doing so, we at least got a result that was useful in practice. Furthermore, we argue that we received insightful information on the usability of the Zachman framework.

When evaluating our study, it should be remembered that it is based on one case and, as such, strong generalizations cannot be made. Our work was planned as a preliminary study of the applicability of the Zachman framework in a small EA-oriented project environment. The Zachman framework deserves more research consideration than can currently be found in the publications of the IS field. We have attempted to contribute to this issue with our small scale study by revealing questions for further lines of enquiry.

For the particular attention of researchers we summarize the following implications from our study.

The Zachman framework is well-known among both practitioners and researchers. However, it was surprising how few analytical results on applying the framework can be found. During our study, we became convinced that more dialogue is needed between the theoretical world and the practical world. Sometimes, valuable tools like the Zachman framework exist somewhere in-between. The problem is, then, that the researchers do refer to such frameworks but do not thoroughly analyze them in their studies, and, in addition, the practitioners recognize them but do not use them.

The message that we wish to communicate is that researchers should be encouraged to carry out studies where the intuitive frameworks presented in literature would be applied in practice in such a way that the experiences would be systematically and critically gathered and analyzed. This development is especially necessary with frameworks that have gained the status of a de-facto standard in a certain area. The status does not mean that there would not be any place for improvement: the danger remains that the status may inhibit a fruitful critical debate.

For practitioners we can pick up the following implications.

In practice, there are often a variety of in-house methods in use. They can be very extensive, like the method in our case study, and applying them in client projects can be difficult. In these practical situations, using a general framework as a benchmarking tool may be an efficient way to modify the method. In our case, the modification meant distilling the most relevant features of the large method.

Modifying a method to meet the practical needs can improve not only the quality and efficiency of an 'ordinary' user of the method but, also, assist a newcomer in becoming quickly acquainted with the method. An adapted version of a method may also help to discuss issues with customers and so forth. As noted by Ylimäki and Halttunen [42], communication plays an important role in EA management. Tools like the lightweight framework are one possibility for enhancing communication, by providing the big picture and boundaries of EA.

5. Further research

In our study, we have applied the Zachman framework in a context where an extensive in-house method is to be used in small EA-oriented projects. We admit that our single case is too limited to make strong generalizations. A single case can, nevertheless, start a discussion by providing information about the problems or weaknesses in the current knowledge. In our case, we have noted that utilizing the well-known Zachman framework raises a few questions.

First, there is the issue of the inadequate definition of the Zachman framework cells, as well as the insufficient analysis of the interconnections between the cells. We can ask how well defined the rows and columns of the Zachman framework really are. They seem intuitively acceptable, but it is hard to see the content of the cells as to be so fixed that they could never be modified. We even argue that the meta-structure of the framework should change over time, although the whole remains the same. Even Zachman himself admits that changes have been made. For example, he has changed the cell example of column 2, row 2 from "Functional Flow Diagram" to "Business Process Model" to better reflect the industry terminology [47]. Zachman argues that the model did not change, but we disagree: it is not just the names of the models that change but also the emphasis of the content, for example, "function" vs. "process".

Second, there is the issue of the complexity of applying the framework. Since the Zachman Framework is not a methodology, a method is needed to fill in the framework cells. While organizations possess and use a variety of methods, there should be further research on how the Zachman framework relates to these in-house methods. More information is needed on the real applicability and usability of the framework in different contexts. To briefly state our conclusions, we believe that (1) better guidelines, (2) analyzed examples, or (3) studies on the adaptability of the Zachman framework would be very useful when applying the framework in practice. The same can be said about many other general frameworks that seem to be valid but remain without real validation by scientific research. Such results would benefit not only the theorists, but they would also make it easier for practitioners to adopt theoretical frameworks in practice.

Third, the application of the Zachman framework in other EA activities, such as in EA quality management or in EA assessment, should be considered in further research.

Finally, the practical work will continue by testing and evaluating the accuracy and usability of the lightweight framework generated in the case study in order to ensure that the use of the lightweight framework will result in the desired effects, both in terms of a more efficient design process and the quality of the outcome. We need to study whether or not the results are better than we would have received from the Zachman framework or the in-house method alone. Additionally, more experience should be obtained from the usage of the lightweight framework, and appropriate metrics should be gathered on its application.

Acknowledgements

The study was conducted as part of a three-year research project called Larkki. It was orchestrated by the Information Technology Research Institute (ITRI), University of Jyväskylä, Finland, and funded by the Finnish National Technology Agency (TEKES) and the participating companies. We wish to thank Kari Koivisto, Risto Kortelainen, Harri Stranden, and Mikko Tulokas at IBM Finland for their valuable time and efforts in this co-operation, as well as Ian Charters, Ian Hogg, and other IBM representatives around the world for their contribution in reviewing this paper.

The material concerning the lightweight framework for small EA-oriented projects has not been submitted to any formal IBM test and is published as is. IBM assumes no responsibility for its accuracy or completeness, and it should not be taken as IBM's official opinion on the fit of the Zachman framework to the IGSM.

References

[1] S. Ambler, *Architecture and Architecture Modeling Techniques,* Agiledata, 2005. Available at http://www.agiledata.org/essays/enterpriseArchitectureTechniques.html.

[2] F.J. Armour and S.H. Kaisler, Enterprise Architecture: Agile Transition and Implementation, *IT Professional* (November/December 2001), 30–37.

[3] F.J. Armour, S.H. Kaisler and S.Y. Liu, A Big Picture. Look at Enterprise Architectures, *IT Professional* (January/February 1999), 35–42.

[4] F.J. Armour, S.H. Kaisler and S.Y. Liu, Building an Enterprise Architecture Step by Step, *IT Professional* (July/August 1999), 31–39.

[5] CIO Council, *Federal Enterprise Architecture Framework* (*FEAF*), Version 1.1, 1999. Available at http://www.cio.gov/documents/fedarch1.pdf.
[6] Department of Defense, *DoD Architecture Framework, Version 1.0, Volume 1, Definitions and Guidelines*, DoD Architecture Framework Working Group, 2004. Available at http://www.defenselink.mil/nii/doc/DoDAF_v1_Volume_I.pdf.
[7] H.-H. Deubler, Employing multiple views to separate large-scale software systems, *The Journal of Systems and Software* **56**(2) (2001), 105–113.
[8] D.J. deVilliers, Using the Zachman Framework to Assess the Rational Unified Process, *The Rational Edge*, March 2001. Available at http://www.therationaledge.com/content/mar_01/t_zachman_dv.thml.
[9] B. Fitzgerald, N.L. Russo and T. O'Kane, Software Development Method Tailoring at Motorola, *Communications of the ACM* **46**(4) (2003), 65–70.
[10] D.S. Frankel, P. Harmon, J. Mukerji, J. Odell, M. Owen, P. Rivitt, M. Rosen and R.M. Soley, The Zachman Framework and the OMG's Model Driven Architecture (white paper). *Business Process Trends*, September 2003. Available at http: //www.bptrends.com/publicationfiles/09%2D03%20WP%20Mapping%20MDA%20to%20Zachman%20Framework%2Epdf.
[11] M. Galic, J. Adams, J.A. Bell, R. Disney, V.-M. Kanerva, S. Matulevich, K. Rebman and P. Spaas, *Patterns: Applying Pattern Approaches. Patterns for e-business Series*, IBM Redbook, 2003.
[12] GERAM, *GERAM: Generalised Enterprise Reference Architecture and Methodology*, Version 1.6.3. IFIP-IFAC Task Force on Architectures for Enterprise Integration, 1999. Available at http://www.cit.gu.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/GERAMv1.6.3.pdf.
[13] F. Goethals, J. Vandenbulcke and W. Lemahieu, *Developing the Extended Enterprise with the FADEE*, Proceedings of the 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, 2004, 1372–1379.
[14] A. Groznik and A. Kovacic, *Business renovation: from business process modelling to information system modelling*, Proceedings of the 24th International Conference on Information Technology Interfaces ITI 2002, June 24–27, 2002, 405–409.
[15] V. Halttunen, A. Lehtinen and R. Nykänen, *Building a Conceptual Skeleton for Enterprise Architecture Specifications*, Proceedings of the 15th European Japanese Conference on Information Modelling and Knowledge Bases, Tallinn, Estonia, May 15–19, 2005.
[16] P. Harmon, Developing an Enterprise Architecture (white paper), *Business Process Trends*, January 2003. Available at http://www.bptrends.com/publicationfiles/Enterprise%20Architecture%20Whitepaper%2D1%2D23%2D03%2Epdf.
[17] D.C. Hay, *Requirements Analysis. From Business Views to Architecture*, Upper Saddle River: Prentice-Hall, 2003.
[18] A. Hirvonen and M. Pulkkinen, A Practical Approach to EA Planning and Development: the EA Management Grid, in: *BIS 2004. Proceedings of 7th International Conference on Business Information Systems*, W. Abramowicz, ed., Poznan, Poland, pp. 284–302.
[19] J. Hoogervorst, Enterprise Architecture: Enabling Integration, Agility and Change, *International Journal of Cooperative Information Systems* **13**(3) (2004), 213–233.
[20] M. Huhns, N. Jacobs, T. Ksiezyk, M. Singh, W.M. Shen, and P. Cannata, *Integrating enterprise information models in Carnot*, Proceedings of International Conference on Intelligent and Cooperative Information Systems, Rotterdam, Holland, 12–14 May 1993, 32–42.
[21] IBM, *Architecture Description Standard: Overview*, version 1.0, IBM Internal Documentation, 13 August 1998.
[22] IBM Global Services, *IBM Global Services Method Release 4.1.1*, IBM Internal Documentation, 2003.
[23] ISO/IEC, ISO/IEC 10746, Reference Model of Open Distributed Processing (RM-ODP), 1995. Available at http://www.dstc.edu.au/Research/Projects/ODP/standards.html.
[24] B. Iyer and R. Gottlieb, The Four-Domain Architecture: An approach to support enterprise architecture design, *IBM Systems Journal* **43**(3) (2004), 587–597.
[25] H. Jonkers, R. van Buuren, F. Arbab, F. de Boer, M. Bonsangue, H. Bosma, H. ter Doest, L. Goroenewegen, J. Guillen Scholten, S. Hoppenbrouwers, M.-E. Iacob, W. Janssen, M. Lankhorst, D. van Leeuwen, E. Proper, A. Stam, L. van der Torre, and G. Veldhuijzen van Zalten, *Towards a Language for Coherent Enterprise Architecture Descriptions*, Proceedings of the Seventh IEEE International Enterprise Distributed Object Computing Conference (EDOC'03), 16–19 September 2003, 28–37.
[26] G. Jucan, A 3D Software Architecture Framework. *The Data Administration Newsletter* (*TDAN.com*), 2004. Available at http://www.tdan.com/i022fe04.htm.
[27] S.H. Kaisler, F.J. Armour and M. Valivullah, *Enterprise Architecting: Critical Problems*, Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS 2005), 2005.
[28] G.P. Kateel, M. Kamath and D.B. Pratt, *An Overview of CIM Enterprise Modeling Methodologies*, Proceedings of the 1996 Winter Simulation Conference, 1996, 1000–1007.
[29] G.R. Khoury and S.J. Simoff, *Enterprise Architecture Modelling Using Elastic Metaphors*, Proceedings of the first Asian-Pacific Conference on Conceptual Modeling, APC CM 2004, (Vol. 31), Dunedin, New Zealand, 2004, 65–69.
[30] M. Lankhorst, *Enterprise Architecture at Work. Modelling, Communication, And Analysis*, Springer-Verlag, 2005.

[31] D. Liles and A. Presley, *Enterprise Modeling Within an Enterprise Engineering Framework,* Proceedings of the 28th Conference on Winter Simulation, Coronado, California, 1996.
[32] J. Morganwalp and A.P. Sage, A System of Systems Focused Enterprise Architecture Framework and Associated Architecture Development Process, *Information Knowledge Systems Management* **3**(2/4) (2003), 87–105.
[33] O. Noran, An Analysis of the Zachman Framework for enterprise architecture from the GERAM perspective, *Annual Reviews in Control* **27** (2003), 163–183.
[34] C. O'Rourke, N. Fishman and W. Selkow, *Enterprise Architecture Using the Zachman Framework,* Thomson Course Technology, 2003.
[35] C.M. Pereira and P. Sousa, *A Method to Define an Enterprise Architecture using the Zachman Framework,* Proceedings of the 2004 ACM Symposium on Applied Computing, 2004, 1366–1371.
[36] A.-W. Scheer, ARIS, in: *Handbook on Architectures of Information Systems,* P. Bernus, K. Mertins and G. Schmidt, eds, Berlin: Springer, 1998, pp. 541–565.
[37] C.A. Singer, Context-specific intellectual capital – The next link in the knowledge chain, *IBM Systems Journal* **42**(3) (2003), 446–461.
[38] J.F. Sowa and J.A. Zachman, Extending and formalizing the framework for information systems architecture, *IBM Systems Journal* **31**(3) (1992), 590–616.
[39] M.W.A. Steen, D.H. Akehurst, H.W.L. ter Doest and M.M. Lankhorst, *Supporting Viewpoint-Oriented Enterprise Architecture,* Proceedings of the 8th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2004), 2004.
[40] The Open Group, *TOGAF Version 8, The Open Group Architecture Framework,* Enterprise Edition, 2002. Available at http://www.opengroup.org/architecture/togaf/.
[41] A. Wegmann, *On the Systemic Enterprise Architecture Methodology* (*SEAM*), Proceedings of the International Conference on Enterprise Information Systems 2003 (ICEIS 2003), Angers, France, 2003.
[42] T. Ylimäki and V. Halttunen, *Perceptions on Architecture Management and Architect's Profession,* Proceedings of the IBIMA 2005 Conference on Information Management in Modern Enterprise, Lisbon, Portugal, July 5–7, 2005.
[43] J.A. Zachman, A framework for information systems architecture, *IBM Systems Journal* **26**(3) (1987), 276–292.
[44] J.A. Zachman, Building the Enterprise – An Infusion of Honesty, *Business Rules Journal* (September 2000). Available at the Business Rules Community Web Site, http://www.brcommunity.com.
[45] J.A. Zachman, Enterprise Architecture Artifacts vs. Application Development Artifacts (Part 1), *Business Rules Journal* (March 2000). Available at the Business Rules Community Web Site, http://www.brcommunity.com.
[46] J.A. Zachman, Enterprise Architecture Artifacts vs. Application Development Artifacts (Part 2), *Business Rules Journal* (May 2000). Available at the Business Rules Community Web Site, http://www.brcommunity.com.
[47] J.A. Zachman, The Zachman Framework and Observations on Methodologies, *Business Rules Journal* (November 2004). Available at the Business Rules Community Web Site, http://www.brcommunity.com.

Tanja Ylimäki is a doctoral student at the Information Technology Research Institute (ITRI), University of Jyväskylä, Finland. She received a degree of Master of Economics in Computer Science and Information Systems from the University of Jyväskylä in 1999. She has worked in several areas including document management, structured documents and metadata. Her current research interests include various aspects of enterprise architecture, such as managing the quality of enterprise architecture.

Veikko Halttunen is a Senior Researcher at the University of Jyväskylä. Previously, he has worked as a Deputy Director of the Information Technology Research Institute (ITRI) at the University of Jyväskylä, as an Associate/Assistant Professor at the University of Jyväskylä and as a Researcher at the Technical Research Centre of Finland. He received his M.Sc. and Ph.Lic. (Information Systems) degrees from the University of Jyväskylä in 1988 and 1997, respectively. His research interests include information systems planning (especially planning methodologies), information system strategies, information architectures, information modeling, and meta-modeling.